Feedback on V4 directions

Nick Rees

diamond

Outline

- Observations from ICALEPCS 2006
- Observations about the current state of EPICS
- Observations about EPICS Version 4
- Suggestions

ICALEPCS 2006 observations

- In past years there was a lot of EPICS
- This year, there was less EPICS, but everyone else had a "framework"
  - CERN had:
    - PVSS-2: A commercial SCADA system
    - OPC: Trying to get it to run on Linux
    - UNICOS: Cryogenic control (PLC/CANbus level).
    - JAPC: Java API for Parameter Control
    - JCOP: Joint COntrols Project
  - ALMA had ACS (ALMA Common Software – based on a software kernel developed by Cosylab for ANKA)
  - ESRF, Soleil, Elettra and Alba had Tango
  - NIFS had ICCS
  - Steve Wampler from ATST was pushing "Middleware Neutral" framework (i.e. abstract the communication layer)

diamond

ICALEPCS 2006 observations

- **Everyone pushed a “three tier architecture”, but there wasn’t any consensus on what the three tiers were…**
- **Overall:**
  - **There was a lot of investment in infrastructure software.**
  - **A lot of it was very good, there was maybe too much.**
  - **A lot of it went far beyond EPICS in overall system integration**
  - **RDB’s were ubiquitous and integrated into the system**

diamond

EPICS observations – the developers

- Core development focuses on the IOC, with little regard for clients and development tools.
- Many, if not most, new requirements are on the client side.
- Client development is fragmented, repetitive and follows no standards.
- All configuration files are fixed format and not extensible – we haven't learnt from the WWW and XML.
- There is a need for better development and debugging tools
- Core development is very conservative
  - Limited adoption of new technologies.
  - Unbundling leads to focusing on a smaller and smaller part of the entire problem.
  - Core development utilises a limited skill-set

EPICS observations – the community

- **The EPICS community is one of the largest in our field, and so should be an asset.**
- **We must respect them and keep them on board**
  - **Backwards compatibility is important!**
  - **EPICS meetings should be more two way streets, with discussion forums.**
- **However, it must be more responsive.**
  - **I sent an email to specific individuals at 15 of the largest sites asking about VDCT, and got only one response.**
  - **There has been very few V4 use cases (mea culpa).**
  - **People claim to be too busy to think strategically.**

EPICS observations – the managers

- A few managers buy into EPICS on the basis that it eliminates the need to develop any software.
- Most managers buy into EPICS on the basis that they don't need to develop infrastructure software.
- Some managers seem to buy in and then take EPICS off in an orthogonal direction to everyone else.
- This is a management failure and we need to do something about it.

EPICS observations – a summary

- A few years ago the EPICS community was widely envied.
- We are rapidly being overtaken by other collaborations who are investing heavily in infrastructure.
- We have problems co-ordinating infrastructure development unless it is done at APS, because of the difficulties in doing large, distributed development.
- EPICS collaboration meetings have become progress reports, not requests for input.
  - We need an open requirements gathering/way forward/feedback forum in every meeting.
- We need better coordination and steady direction.
- Unless we solve these problems EPICS will die.

diamond

EPICS Version 4

- EPICS version 4 was a laudable attempt to kick-start EPICS on an aggressive development direction.
- It focussed virtually entirely on the IOC, and despite some brave attempts by some people (give me all the BPM's!), it virtually ignored high level requirements.
- The development team worked very democratically.
- The requirements seemed reasonable, but there was limited buy-in from the community.
- The arguments seems to be all about software nerd issues, and not what was really wanted – either by the EPICS community or the science requirements.
- It had all the hallmarks of becoming very clever software that couldn't be sold because no-one needed it.

EPICS Version 4 requirements

- **Most of these seemed reasonable. Who can argue with things like:**
  - **Name introspection**
  - **Removal of string length restrictions**
  - **Triggers and filters**
- **I personally, could not get my head around the implications of arbitrarily complex heirarchies for developers, without more powerful development tools.**
- **On the other hand, I liked arrays of links, and the associated link behaviours (block/wait etc).**

EPICS Version 4 was not EPICS V3!

- It had incompatible wire protocol, incompatible file formats, possibly incompatible API's.
- It could be said that EPICS V4 was more akin to LabView or a SCADA system than EPICS V3.
- Some things were treated too lightly
  - It seems to be to be very difficult to write a perfect gateway.
  - The slow take-up of R3.14 was mainly due to the change in the build system. How slow would the V4 take-up be?
  - It was unclear how to build complex systems in the new paradigm, without good development tools.
  - It was not clear to me what these development tools would look like.

diamond

How to run a successful project

- There must be a “killer app”
  - This means that the merits of the new system must be so invaluable to someone that funding is assured.
  - This normally means tying the development to a new project that requires new functionality.
- You need an architect
  - Someone who is the ultimate arbiter of any technical decisions.
- You need a customer
  - Someone who represents the end users and is the ultimate arbiter of the priority of requirements.
- You need a manager
  - Someone who makes sure the other two remember the cost and time implications.
- EPICS V4 had none of these - democracies don’t work in software development!

diamond

Suggestions

- **Backwards compatibility is a primary requirement.**
- **Create the development tools hand in hand with the core developments.**
- **Tie the development to something that needs it and so really sets the requirements.**
- **Focus the strategic development in the high-level area – the low level is already good, and can probably be progressed in a series of small focussed improvements.**
- **Managers are failing – we must work out how to stabilise the funding, coordination and direction**

diamond

What am I (or is DLS) doing?

- I am failing along with the rest of us…
- However, we have chosen motion control and VDCT as two strategic areas that are small enough for us to handle and cut out teeth on and significant enough to make a serious impact.
- We are also committed to supporting core development (whatever that is) in some way to the tune of at least 1 additional FTE.
- We advertised last year and got nothing
- We will be advertising in the next few weeks in a second attempt.
- It's a really good place to be, come join us ☺

diamond

Conclusions

- **You're the community – what do you conclude?**